REMARKS

Claims 1-22, 38-46, and 54-74 are currently pending in this application. Claims 2, 12, 39, 44, 55, and 64 are currently amended. Claims 23-37 and 47-53 were previously withdrawn.

Applicant filed an Information Disclosure Statement (IDS) on form PTO-SB-08 on June 12, 2007 and herewith this amendment. Applicant respectfully requests confirmation of the Office's consideration of the submitted information in the next Office Action.

Claim Objections

The Examiner objected to Claims 12, 44, and 64 stating that they "recite acronym 'HDL', such acronym should be spelled out." The Applicants point out the "HDL" is a portion of a proper name for a product in which letters are not spelled out. However, the Applicants have amended Claims 12, 44, and 64 to read in pertinent part "Verilog HDL hardware description language" and believes the objections to these claims are overcome by the amendments.

Claim Rejections - 35 U.S.C. §112

Claims 2, 39, and 55 are rejected under 35 U.S.C. §112, second paragraph as being indefinite because the limitation "the length" in each claim has insufficient antecedent basis. Applicants respectfully disagree with the Examiner's rejection. However, without conceding the lack of antecedent basis, and in the interests of advancing the application to allowance, the Applicants have amended Claims 2, 39, and 55 to recite "a length." The Applicants believe the rejections of Claims 2, 39, and 55 are overcome by the amendments.

Claim Rejections - 35 U.S.C. §103

Claims 1-22, 38-46, and 54-74 stand rejected under U.S.C. § 103(a) as being unpatentable over Moberg et al. (US 6,698,015 B1) in view of Dice et al. (US 6,788,236 B1). The Applicants respectfully traverse the Examiner's rejections.

Claim 1

The present invention is directed to a method for modifying a program to allow the program to execute on a processor system that includes a programmable logic device. Moberg, on the other hand, does not involve having certain functions performed by a PLD, but only "performance improvement of critical code execution using shared libraries and/or cache locking techniques." (Moberg, Col. 1:15-16.)

In the first instance, the Examiner's argument that Moberg shows "replacing the critical code segment with a statement that calls the function," (Office action p. 3.) is misplaced. As the Examiner admits, Moberg never rewrites a critical code segment as a function (Office Action p. 4.) but merely selects critical functions, compiles them, and stores them separately. The Examiner is thus assuming that the critical code has already been rewritten. Nothing in Mober teaches or suggests such rewriting, or the applicability of the invention described therein to such written code. Since the compiled portions are themselves functions, they are not replaced with statements that call functions.

The Examiner next asserts that Dice shows rewriting critical code as functions. "Dice discloses operation performing the reset operation includes replacing current processor information existing within the current state of the critical code with the processor information obtained from the registered or saved state of the critical code (col. 8:1-10)." (Office Action p. 4.) The Applicants respectfully submit that the Examiner is in error.

Dice teaches a method for handling interrupts. Dice does not teach "rewriting the critical code segment as a function" either before or after the interrupts. Critical code is defined in Dice as "a series of one or more logic instructions . . . which a processor in the computer system must execute from start to finish without any interference." (Dice, Col. 2:1-5.) Critical code is registered:

> "with an operating system for invocation of a critical execution manager in the event of an interruption to the critical code. The registration process can cause the operating system to store the address bounds of the critical code in a data structure for use by the critical execution manager upon returning from handling an interruption to execution of the critical code." (Dice, Col. 6:65 – 7:4.)

Thus, the registered information includes only information about the critical code preserved before servicing interrupts, but not a critical code segment or a function.

Rewriting state information after interrupts is not rewriting critical code. Typically, interrupt handling includes saving the state of the processor including various registers and the program counter. Upon returning from the interruption, the registers are restored and the program counter is updated with the saved value to continue execution where the processor left off. In Dice, by contrast, after an interrupt occurs during execution of critical code, a critical execution manager restarts the critical code over rather than continue where it left off. The critical execution manager uses the state information registered for the critical code, instead of the current value in the program counter that was restored by the interrupt handling routine, to determine where execution is to begin upon return from the interruption. Thus, it is the saved state (registered information) of the critical code that is written to the processor registers and program counter to replace the current state of the critical code, not the critical code segment itself:

> Generally, the operation of performing the reset operation includes replacing current processor information (e.g., a current

> program counter) existing within the current state of the critical code with processor information obtained from the registered or saved state of the critical code (or of the registered thread) in order to allow the operation of returning to execution of the critical code to begin execution (e.g., restart) of the critical code without interference from handling the interruption. (Dice, Col. 8:2-10.) (emphasis added)

Moreover, the saved state and the current state of the critical code include only processor specific information, not the code itself, and certainly not "the critical code segment as a function" as recited in Claim 1. Dice states:

> According to embodiments of the invention, the state of critical code (i.e., the save state and/or a current state) can include information including processor information related to the critical code such as register values, flag settings or other processor specific information. The state of critical code can also include memory information that is related to the critical code such as addresses and/or values of memory locations referenced by instructions within the critical code. (Dice, Col. 8:16-23.) (emphasis added)

In addition, since neither Moberg nor Dice teach or suggest "rewriting the critical code segment as a function" or "replacing the critical code segment with a statement that calls the function," as recited in Claim 1, even combining them does not teach all the elements of Claim 1. For at least the above reasons, the Applicants believe that Claim 1 and Claims 2-22 which depend therefrom are allowable.

Claim 38

The Examiner asserts that "per claim 38, this is another method article version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above." (Office Action p. 6.) The Applicants respectfully submit that all the claim limitations in Claim 38 have not been addressed or covered by the Examiner's rejection of Claim 1.

First, the Applicants respectfully point out that Claim 38 includes an additional limitation not found in Claim 1 which is therefore not addressed or covered in the Examiner's remarks with respect to Claim 1. Claim 38 recites in part "replacing the critical code segments with at least one extended instruction, not included in the native instruction set of the processor." Claim 1 does not include the limitations of an "extended instruction, not included in the native instruction set of the processor." There is no teaching or suggestion whatsoever in either Moberg or Dice regarding such extended instructions. For at least this reason, the Applicants respectfully request the Examiner to withdraw the rejection and allow Claim 38.

Further, with respect to steps in Claim 38 common or similar to those in Claim 1, the Applicants repeat the arguments set forth above with respect to Claim 1. For at least the above reasons, the Applicants believe that Claim 38 and Claims 39-46 which depend therefrom are allowable.

Claim 54

The Examiner asserts that "per claim 54, this is [a] system version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above." (Office Action p. 7.) The Applicants repeat the arguments set forth above with respect to Claim 1. For at least the above reasons, the Applicants believe that Claim 54 and Claims 55-73 which depend therefrom are allowable.

Claim 74

The Examiner asserts that "per claim 74, this is the method version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above." (Office Action p. 10. The Applicants respectfully submit that all the claim limitations in Claim 74 have not been addressed or covered by the Examiner's rejection of Claim 1.

First, the Applicants respectfully point out that Claim 74 includes an additional limitation not found in Claim 1 which is therefore not addressed or covered in the Examiner's remarks with respect to Claim 1. Claim 1 does not include the limitations of "replacing the critical code segment with one or more extended instructions; and compiling the revised program such that the extended instructions are executed by the programmable logic device," as recited in Claim 74. There is no teaching or suggestion whatsoever in either Moberg or Dice regarding such extended instructions or executing such extended instructions by a programmable logic device. For at least these reasons, the Applicants respectfully request the Examiner to withdraw the rejection and allow Claim 74.

Second, with respect to steps in Claim 74 common or similar to those in Claim 1, the Applicants repeat the arguments set forth above with respect to Claim 1. For at least the above reasons, the Applicants believe that Claim 74 is allowable.

The Applicants believe that all pending claims are allowable and respectfully request that the Examiner issue a Notice of Allowance. Should the Examiner have questions, the Applicant's undersigned representative may be reached at the number provided below.

Respectfully submitted,

Kenneth M. Williams et al.

Date: September 10, 2007

Ronald Rohde, Reg. No. 45,050

Carr & Ferrell *LLP*
2200 Geng Rd.
Palo Alto, CA 94303
Phone (650) 812-3487
Fax (650) 812-3444